Forecasting the Average Monthly Temperatures in Los Angeles, USA

Yongheng Zan

Abstract

The main point of my point is to predict the temperature in the coming months by analyzing the average monthly temperature in Los Angeles over the past few decades. I used data cleaning, data transform, identify model, diagnostic check and compare test and true value to help me process this project. I'm successful achieve my goals and predict the values at the end.

Introduction

I chose the data set from the Kaggle.com which includes the monthly average temperature of Los Angeles. With the development of society, our human activities are inextricably linked to the rise of global temperature. I built the model in LA to help us see trends and determine if average temperatures are really rising. I think it is very important to learn and apply time series analysis. In addition, similar analyses have the potential to be applied on a larger scale and with greater impact.

I was following the following techniques to help me achieve my goal. Firstly, I divide the data into training data and test data to check whether my prediction is correct later. Secondly, I used data transformation and construct model to predict the data. Then, I use diagnostic check to see if my model is reasonable and suitable. Finally, comparing the predicted data with real data to verify my model.

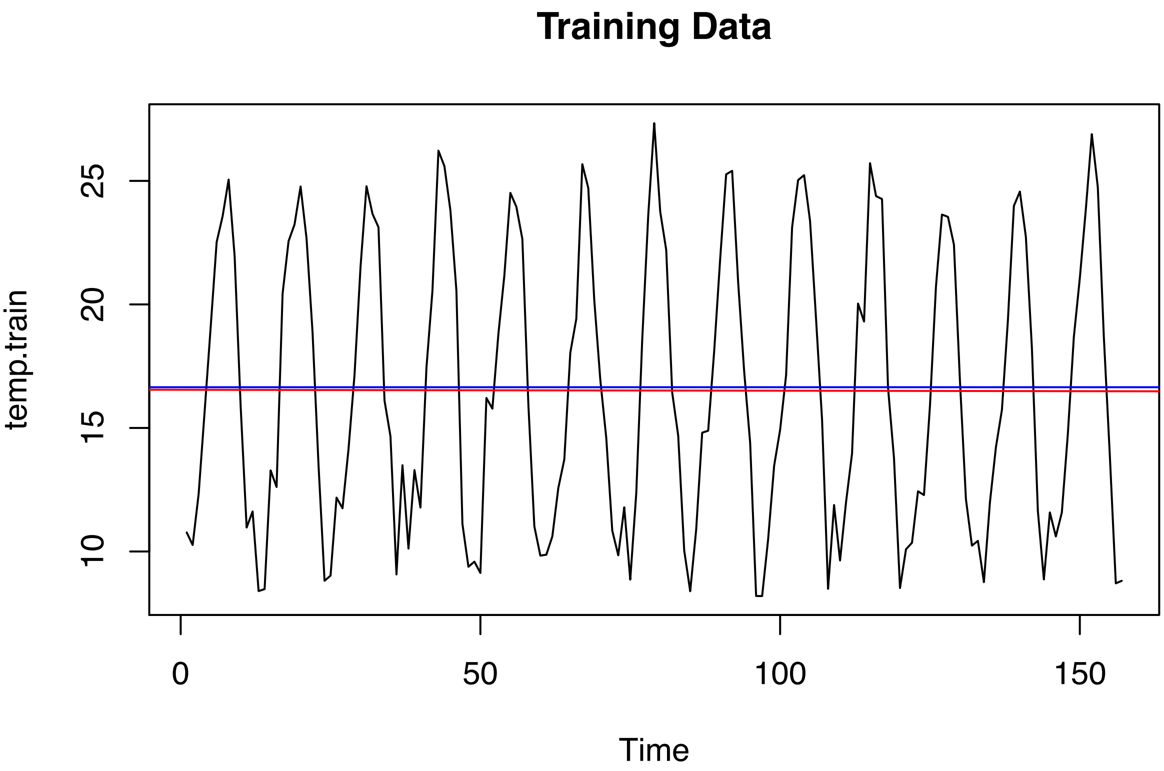
For the result, I can clearly see my prediction is successful by comparing the predict and real value that the predict values are likely the real values and fall in the 95% confident interval.

I used the RStudio to process this project, and all resource listed at the end.

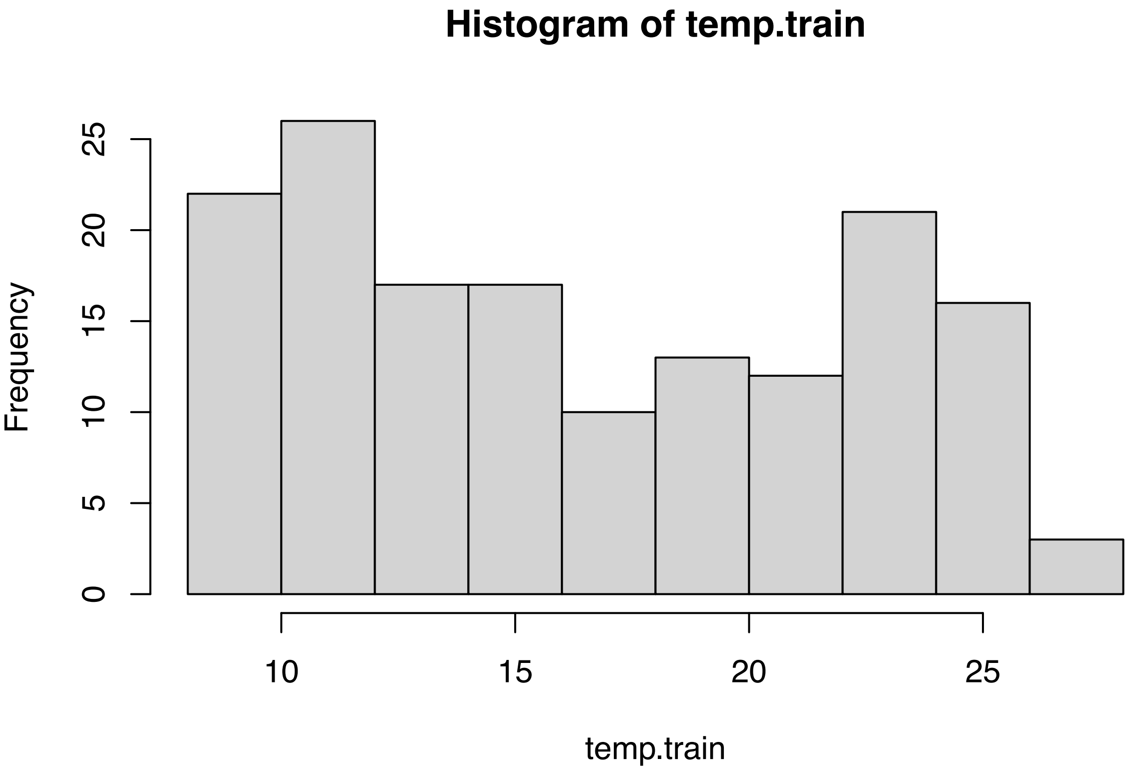
Part 1. Plot and analyze the time series

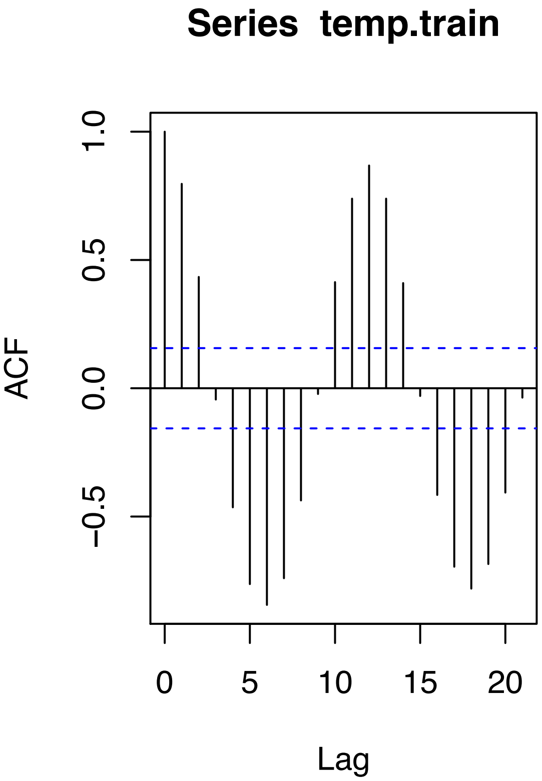
1.Plot the Training Data

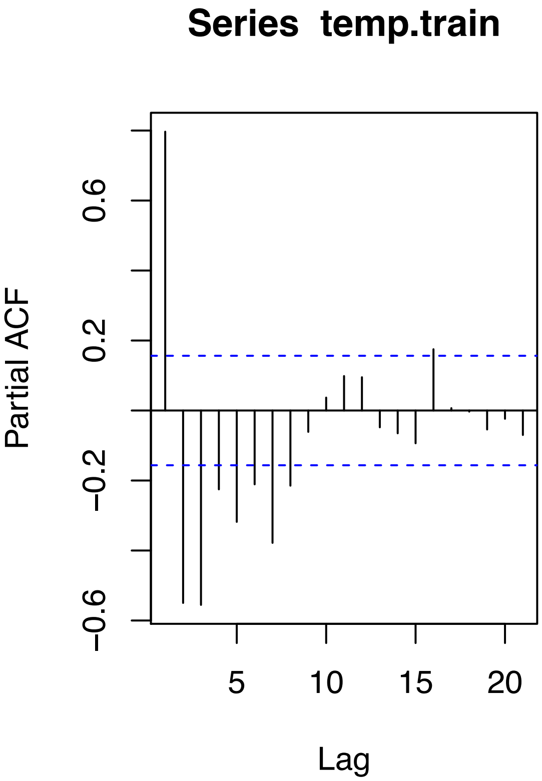
```
# temp.train
plot.ts(temp.train, main='Training Data')
# add trend line and mean line
fit <- lm(temp.train~as.numeric(1:length(temp.train)))
abline(fit,col='red')
abline(h=mean(temp_raw$AverageTemperature), col='blue')
```



Obvervation:
a. From the graph we can see that the plot of training data doesn't has a obvious trend.
b. The data set has seasonality which is right because we are analysis the temperature during year by year. And it isn't appear stationary and we will explore later.
c. The temperature variation remains roughly same over time for the most part and there aren't any apparent sharp changes in behavior besides seasonal effect.







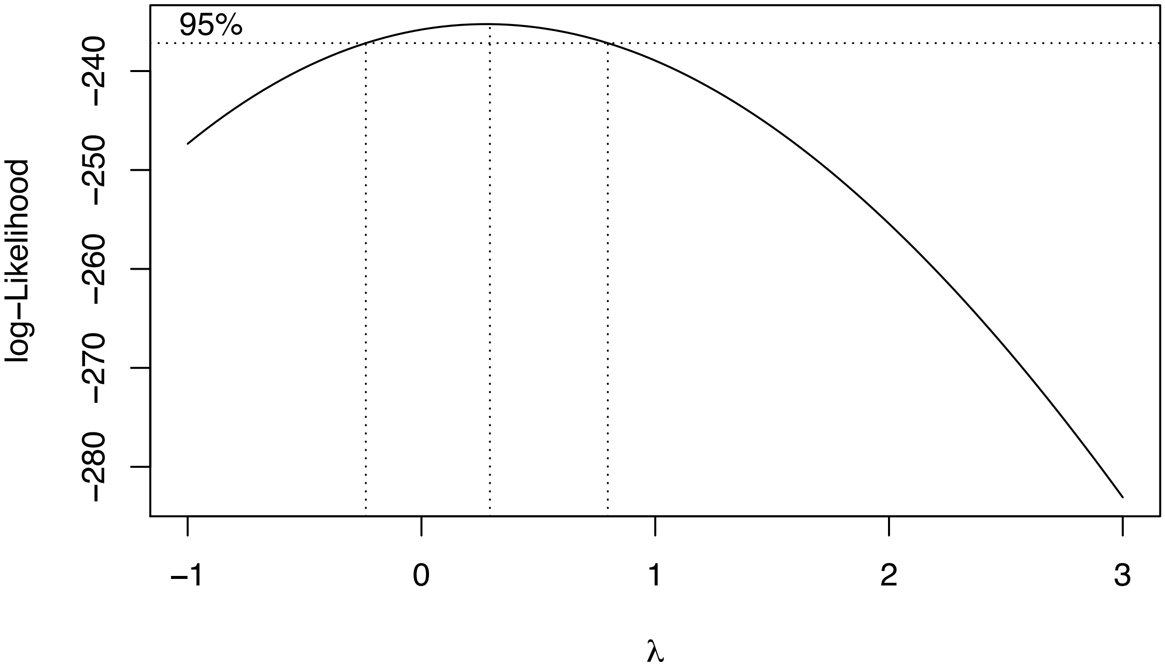
From the histogram, we can notice that the data is nearly normal but has left-skew shape, and we can see the data is not stationary from the ACF plot but all spikes seemly repeat which may due to the seasonality.

2.Perform box-cox transformation on training data

```
var(temp.train)
```

```
## [1] 32.39398
```

```
# using Box transformation
bcTransform <- boxcox(temp.train ~ as.numeric(1:length(temp.train)),
                      lambda=seq(-1,3),plotit=TRUE)
```



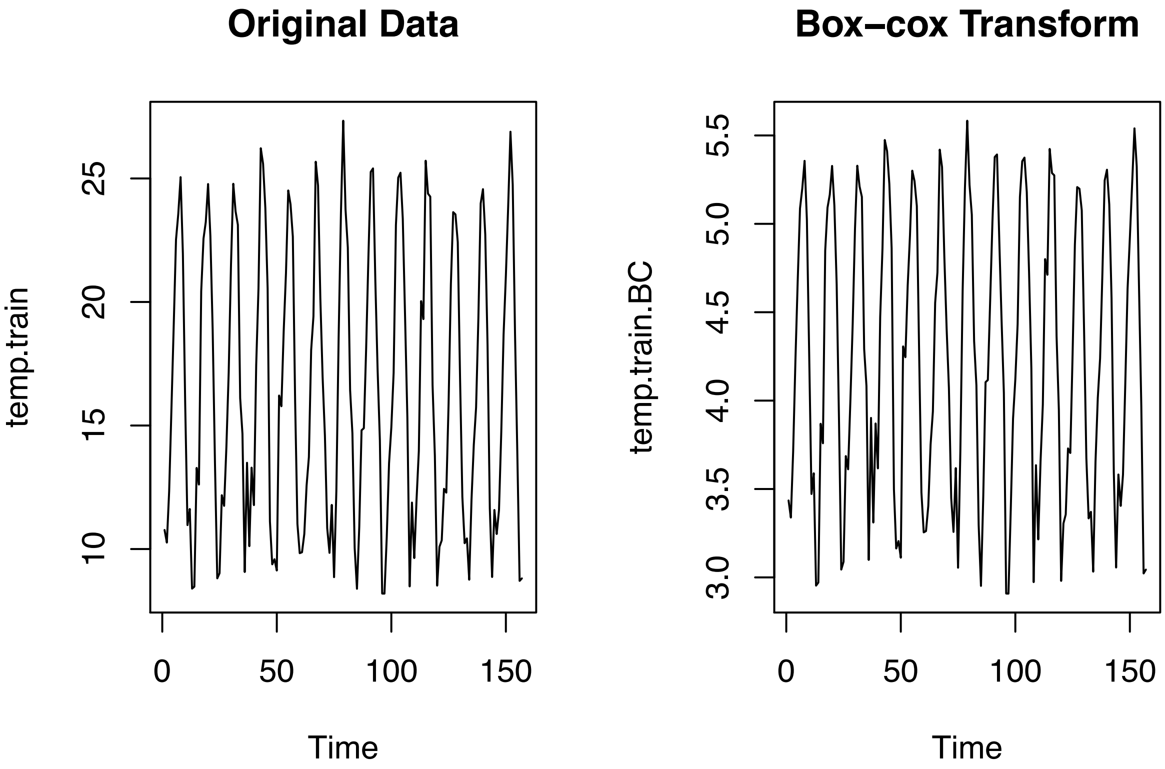
```
lambda <- bcTransform$x[which.max(bcTransform$y)]
lambda
```

```
## [1] 0.2929293
```

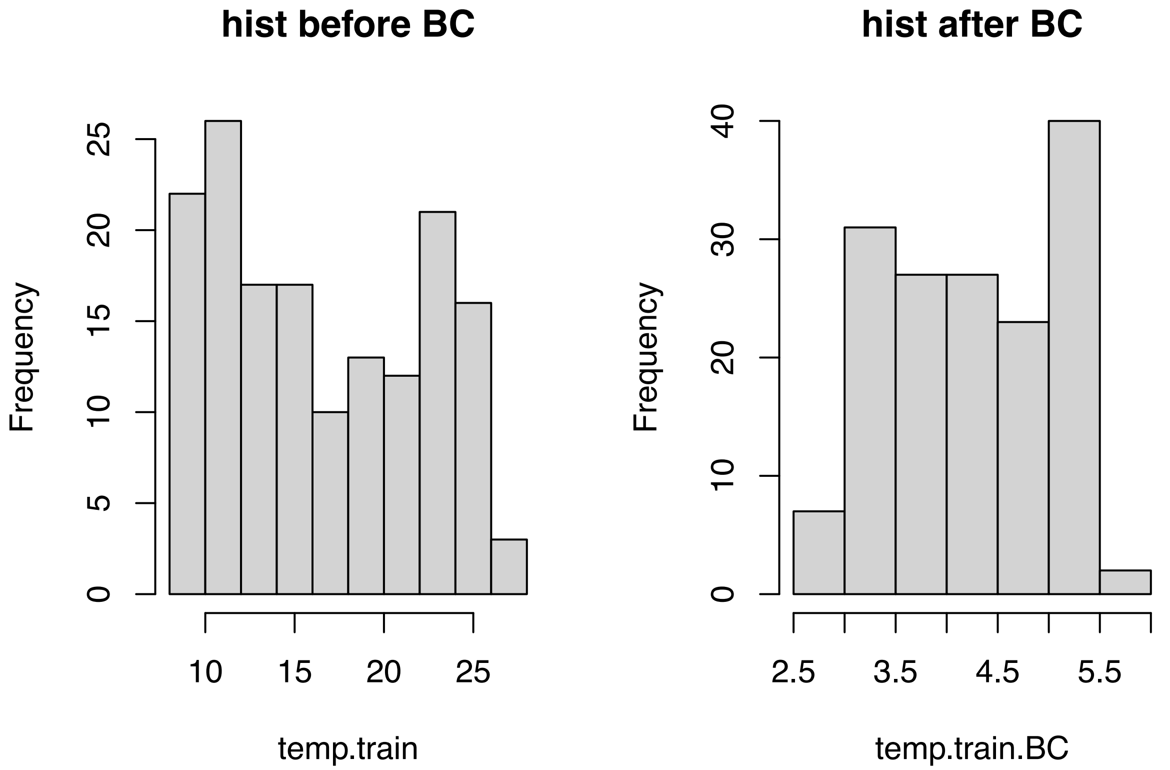
```
temp.train.BC <- (1/lambda)*(temp.train^lambda-1)
```

The data set has big variance, so we are going to use Box-cox transformation to decrease the variance of data. And compare the training data with transform data.

```r
# compare original and box-cox data
par(mfrow=c(1,2))
plot.ts(temp.train,main='Original Data')
ts.plot(temp.train.BC, main = "Box-cox Transform")
```



```r
# compare the normal distribution
par(mfrow=c(1,2))
hist(temp.train, main = 'hist before BC')
hist(temp.train.BC,main = 'hist after BC')
```



```
# compare variance
var(temp.train)
```

```
## [1] 32.39398
```

```
var(temp.train.BC)
```

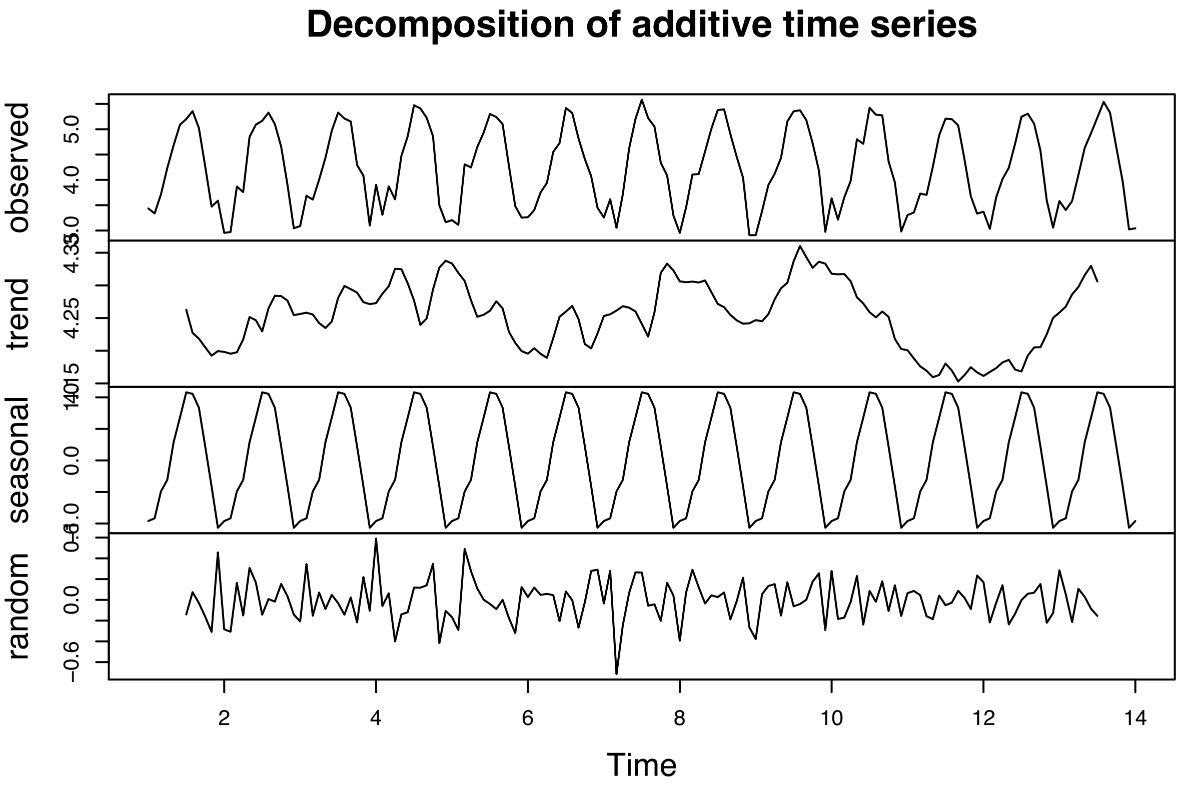
```
## [1] 0.6397146
```

We can see that both time series plot roughly same, and the transform histogram is more normal and symmetric. Furthermore, the variance drop huge.

Part 2. Differencing data

1.Decomposition of Box-Cox Transformed Data

```
# decomposition of BC transformed data
temp_new <- ts(temp.train.BC,frequency = 12)
decomp <- decompose(temp_new)
plot(decomp)
```

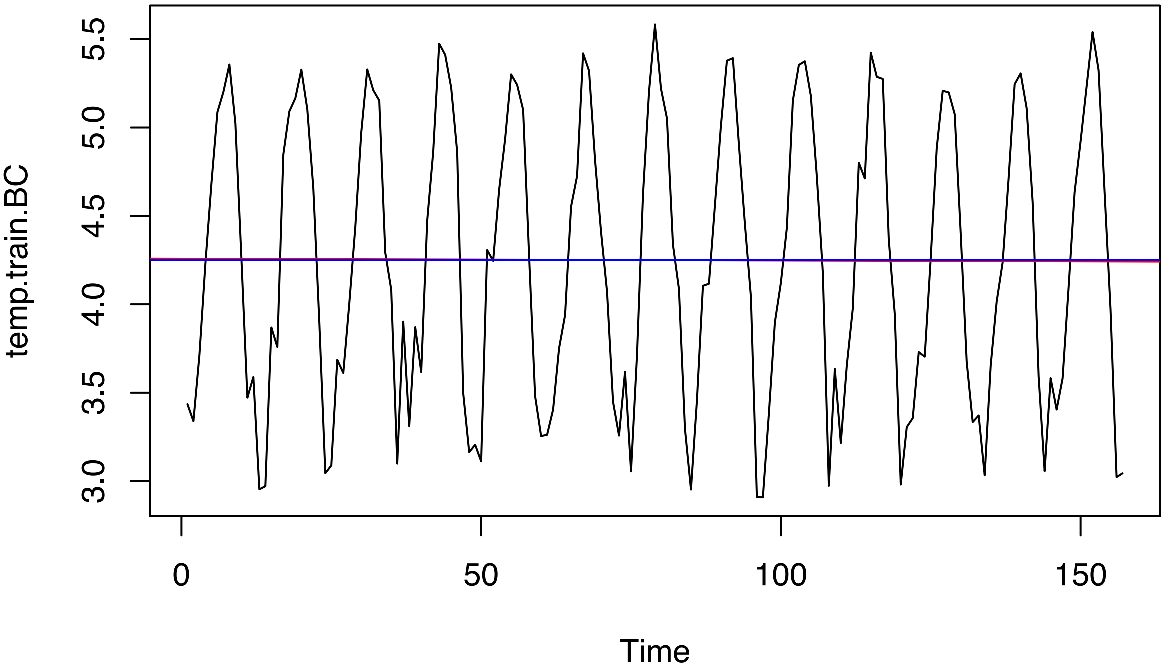


The decomposition shows us a trend and seasonal, then we are going to difference lags to remove trend and seasonal.

2.Differencing data process

No differencing

```
# No differencing
fit1<-lm(temp.train.BC~as.numeric(1:length(temp.train.BC)));
plot.ts(temp.train.BC)
abline(reg=fit1,col='red')
abline(h=mean(temp.train.BC),col='blue')
```

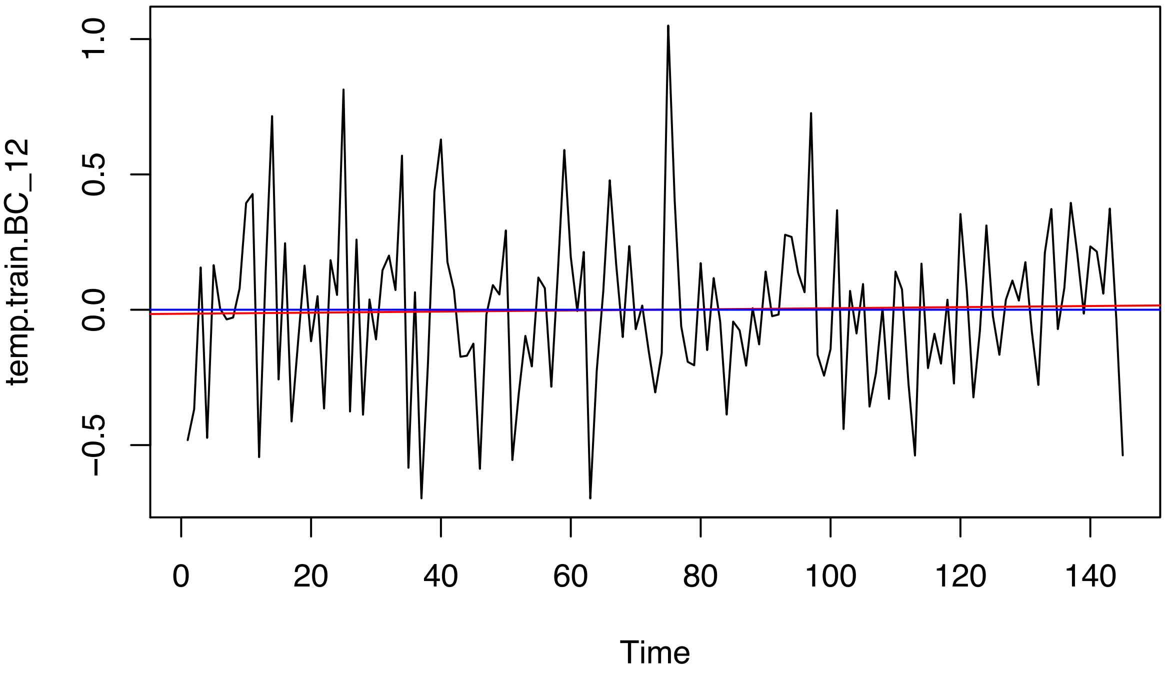


```
##   Differencing Variance
## A None         0.639714635713525
## B
## C
```

There has seasonal component and may contains a trend, so we start different at 12 first.

Difference at lag 12

```
# difference at 12 to remove the seasonality
temp.train.BC_12 <- diff(temp.train.BC,lag=12)
plot.ts(temp.train.BC_12)
fit3<-lm(temp.train.BC_12~as.numeric(1:length(temp.train.BC_12)))
abline(reg=fit3,col='red')
abline(h=mean(temp.train.BC_12),col='blue')
```

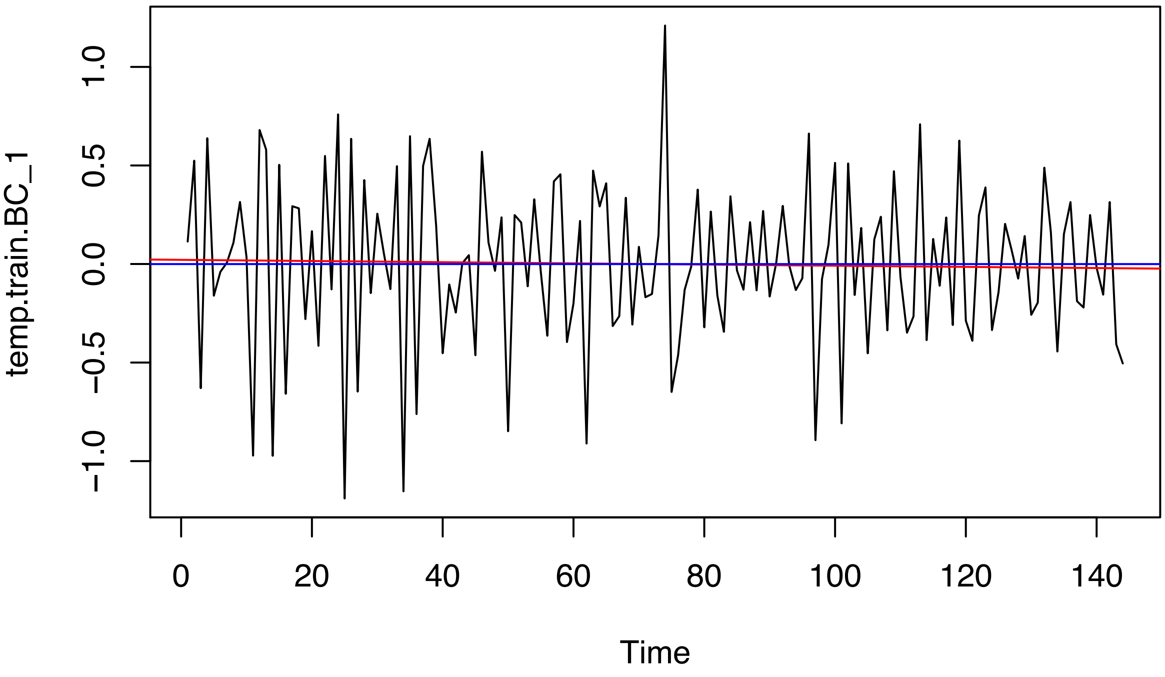


```
##   Differencing Variance
## A None         0.639714635713525
## B lag_12       0.0895920871212595
## C
```

The seasonal component has been removed and the variance decrease as well which mean we are in the right direction. Furthermore, the plot shows us it nearly stationary but still has a slightly trend so we are going to difference at 1 in the next step.

Dfference at lag 1

```
# difference at 1
temp.train.BC_1 <- diff(temp.train.BC_12,lag=1)
plot.ts(temp.train.BC_1)
fit4<-lm(temp.train.BC_1~as.numeric(1:length(temp.train.BC_1)))
abline(reg=fit4,col='red')
abline(h=mean(temp.train.BC_1),col='blue')
```

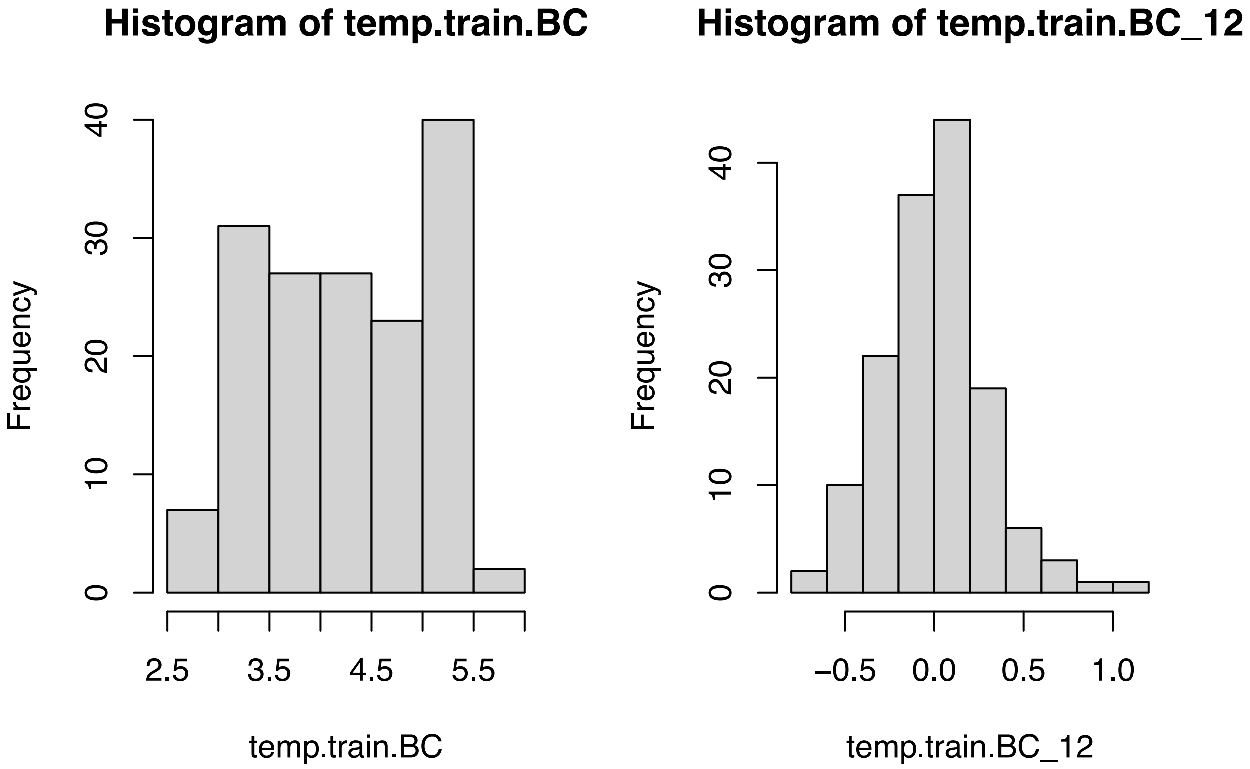


```
##   Differencing Variance
## A None         0.639714635713525
## B lag_12       0.0895920871212595
## C lag12&1      0.179154621089446
```

Since the variance increase which mean we over differencing, so we discard difference at lag 1.

Compare histogram

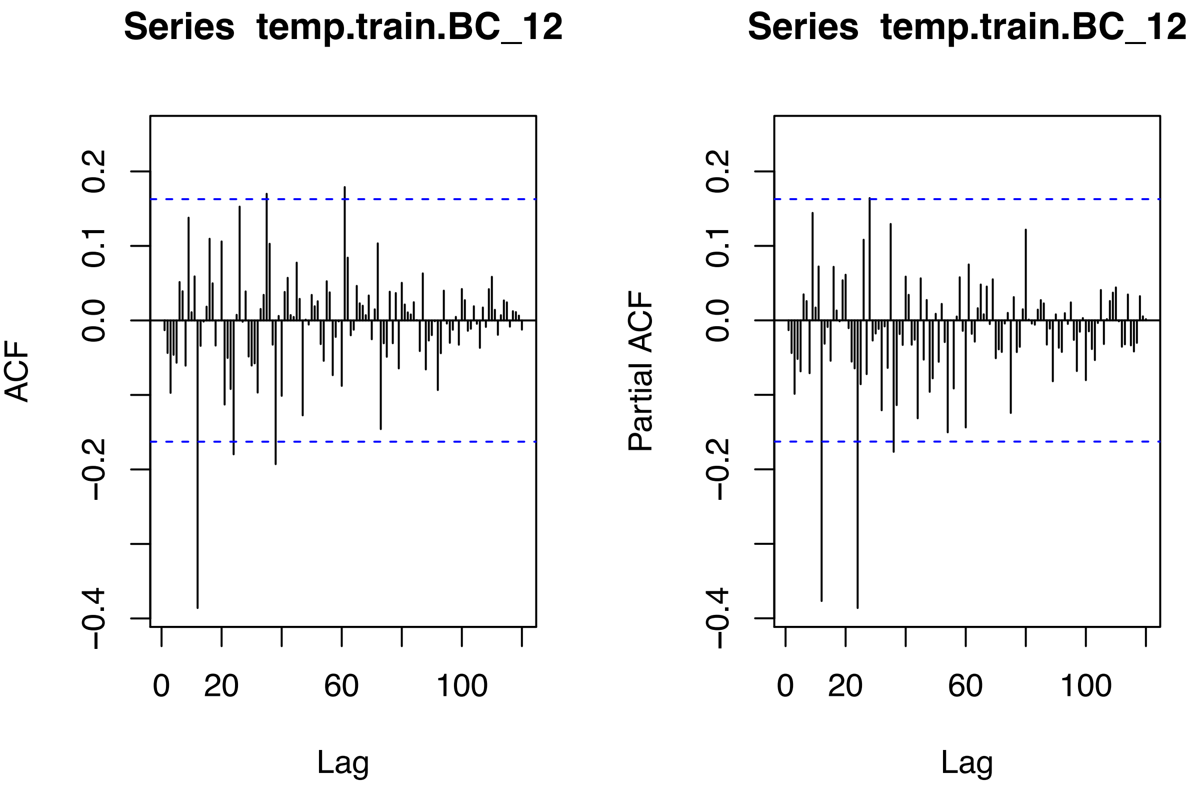
```
# compare his
par(mfrow=c(1,2))
hist(temp.train.BC)
hist(temp.train.BC_12)
```



By comparing the histogram with/out difference at 12, we can see that the histogram with difference at lag 12 seems to be approximately normally distributed.

Part 3. Model identification using ACF/PACF

```
par(mfrow=c(1,2))
Acf(temp.train.BC_12,lag.max=120)
Pacf(temp.train.BC_12,lag.max=120)
```



We have the following parameters base on the ACF and PACF:
s = 12 – SARIMA model
D = 1 – differenting at 12
d = 0 – no differenting at 1
P = 1-2 – there are spikes at 12,24,36 from PACF
Q = 1 – There are spikes at 12 from ACF
q = 0 – which are all within the first period from ACF
p = 0 – which are all within the first period from PACF

Part 4. Comparing AICC and Choosing Candidate Models

We are going to use the AICc() and arima() functions to create some candidate models and compare their AICCs. I'm going to use 'for' loop to achieve the algorithm.

```
df <- expand.grid(P=1:2,Q= 0:2 )
df <- cbind(df, AICc=NA)

# Compute AICc:
for (i in 1:nrow(df)) {
  sarima.obj <- NULL
  try(arima.obj <- arima(temp.train.BC, order=c(0, 0, 0),
                      seasonal=list(order=c(df$P[i], 1, df$Q[i]), period=12),
                      method="ML"))
  if (!is.null(arima.obj)) { df$AICc[i] <- AICc(arima.obj) }
  # print(df[i, ])
}
df[order(df$AICc), ]
```

```
##   P Q       AICc
## 4 2 1 -13.488374
## 6 2 2 -11.391821
## 3 1 1  -9.172111
## 5 1 2  -7.684880
## 2 2 0  11.521432
## 1 1 0  38.699321
```

By comparing the AICc, I'm going to use the first and last two as my candidate models.
Model 1: SARIMA(0,0,0)x(2,1,1) AICc: -13.488374
Model 2: SARIMA(0,0,0)x(2,1,2) AICc: -11.391821
Model 3: SARIMA(0,0,0)x(2,1,0) AICc: 11.521432
Model 4: SARIMA(0,0,0)x(1,1,0) AICc: 38.699321

Part 5. Estimating Coefficients & check Invertibility/Stationaty

Model_1

$$SARIMA(0,0,0)(2,1,1)_{12}$$

```
model_1 <- arima(temp.train.BC,order=c(0,0,0),
                 seasonal=list(order=c(2,1,1),period=12),method="ML")
model_1
```

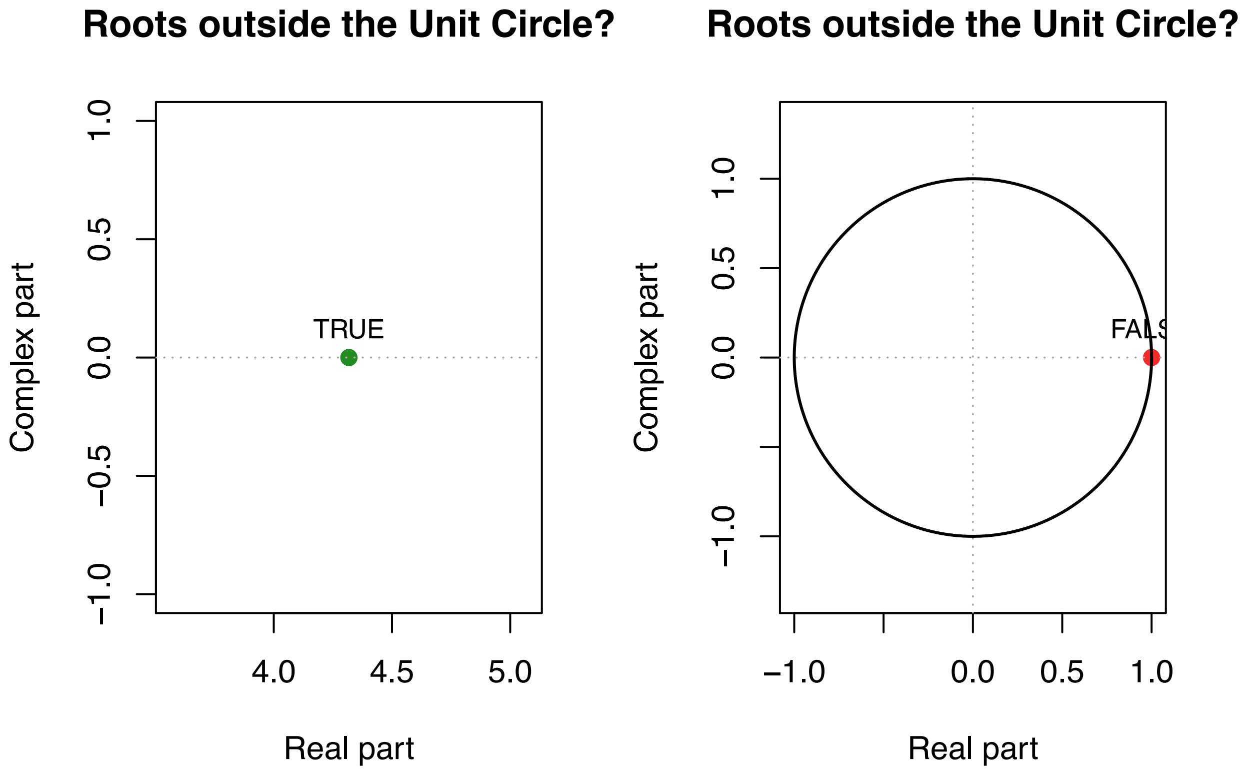
```
##
## Call:
## arima(x = temp.train.BC, order = c(0, 0, 0), seasonal = list(order = c(2, 1,
##     1), period = 12), method = "ML")
##
## Coefficients:
##          sar1     sar2     sma1
##       -0.0784  -0.2316  -1.0000
## s.e.   0.0863   0.0878   0.1796
##
## sigma^2 estimated as 0.03883:  log likelihood = 10.82,  aic = -13.65
```

$$(1+0.2316B^{24})(1-B^{12})X_t = (1-1B^{12})Z_t$$

```
## [1] 4.317789+0i
```

```
##       real complex outside
## 1 4.317789       0    TRUE
## *Results are rounded to 6 digits.
```

```
##   real complex outside
## 1    1       0   FALSE
## *Results are rounded to 6 digits.
```



As we can see, the sar1 could potentially have zero as it coefficient, since includes zero in the interval (+-2 x SE). And by calculating the ploy root of AR part, we find the root is outside the unit circle which imply this model is stationary. However, the moving average part has unit root and isn't outside the unit circle which imply this model is not invertible. So this model is stationary but not invertible.

Model_2

$$SARIMA(0,0,0)(2,1,2)_{12}$$

```
model_2 <- arima(temp.train.BC,order=c(0,0,0),
                 seasonal=list(order=c(2,1,2),period=12),method="ML")
model_2
```

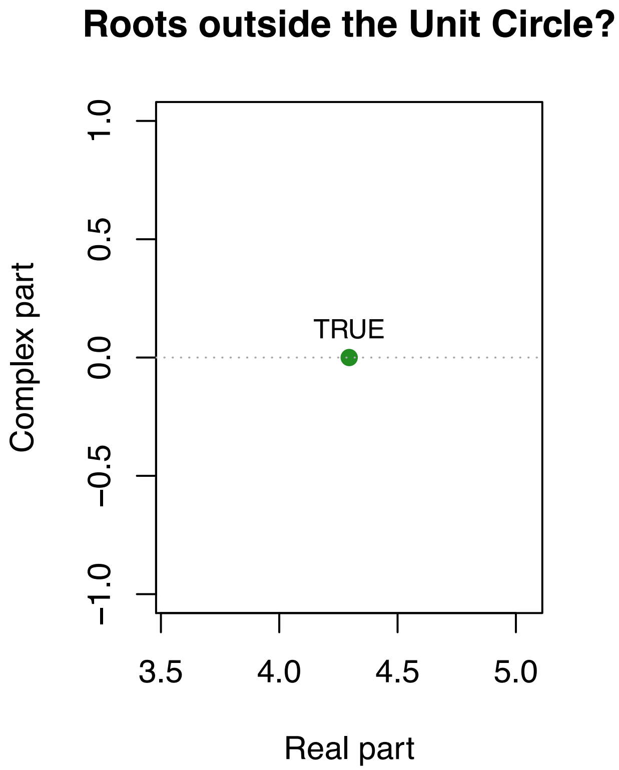
```
##
## Call:
## arima(x = temp.train.BC, order = c(0, 0, 0), seasonal = list(order = c(2, 1,
##     2), period = 12), method = "ML")
##
## Coefficients:
##          sar1     sar2     sma1     sma2
##       -0.1228  -0.2328  -0.9514  -0.0485
## s.e.   0.4665   0.0881   0.5292   0.5009
##
## sigma^2 estimated as 0.0389:  log likelihood = 10.83,  aic = -11.65
```

```
AICc(arima(temp.train.BC,order=c(0,0,0),
                 seasonal=list(order=c(2,1,2),period=12),method="ML"))
```

```
## [1] -11.39182
```

$$(1+0.2328B^{24})(1-B^{12})X_t = Z_t$$

```
##       real complex outside
## 1 4.295533       0    TRUE
## *Results are rounded to 6 digits.
```



As we can see, the sar1, sma1, sma2 could potentially have zero as their coefficient, since includes zero in the interval (+-2 x SE). And by calculating the ploy root of AR part, we find the root is outside the unit circle which imply this model is stationary. And base on this is a pure AR model which is also invertible. So this model is stationary and invertible.

Model_3

$$SARIMA(0,0,0)(2,1,0)_{12}$$

```
model_3 <- arima(temp.train.BC,order=c(0,0,0),
                 seasonal=list(order=c(2,1,0),period=12),method="ML")
model_3
```

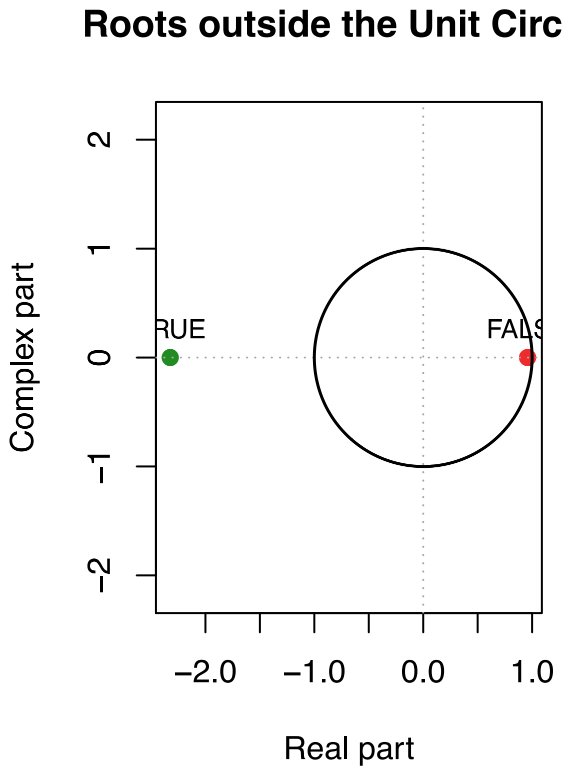
```
##
## Call:
## arima(x = temp.train.BC, order = c(0, 0, 0), seasonal = list(order = c(2, 1,
##     0), period = 12), method = "ML")
##
## Coefficients:
##          sar1     sar2
##       -0.6130  -0.4491
## s.e.   0.0777   0.0753
##
## sigma^2 estimated as 0.05762:  log likelihood = -2.72,  aic = 11.44
```

```
AICc(arima(temp.train.BC,order=c(0,0,0),
                 seasonal=list(order=c(2,1,0),period=12),method="ML"))
```

```
## [1] 11.52143
```

$$(1+0.613B^{12}+0.4491B^{24})(1-B^{12})X_t = Z_t$$

```
##        real complex outside
## 1  0.958392       0   FALSE
## 2 -2.323344       0    TRUE
## *Results are rounded to 6 digits.
```



As we can see, by calculating the ploy root of AR part, we find the root isn't outside the unit circle which imply this model is not stationary. And base on this is a pure AR model which is invertible. So this model is not stationary but invertible.

Model_4

$$SARIMA(0,0,0)(1,1,0)_{12}$$

```
model_4 <- arima(temp.train.BC,order=c(0,0,0),
                 seasonal=list(order=c(1,1,0),period=12),method="ML")
model_4
```

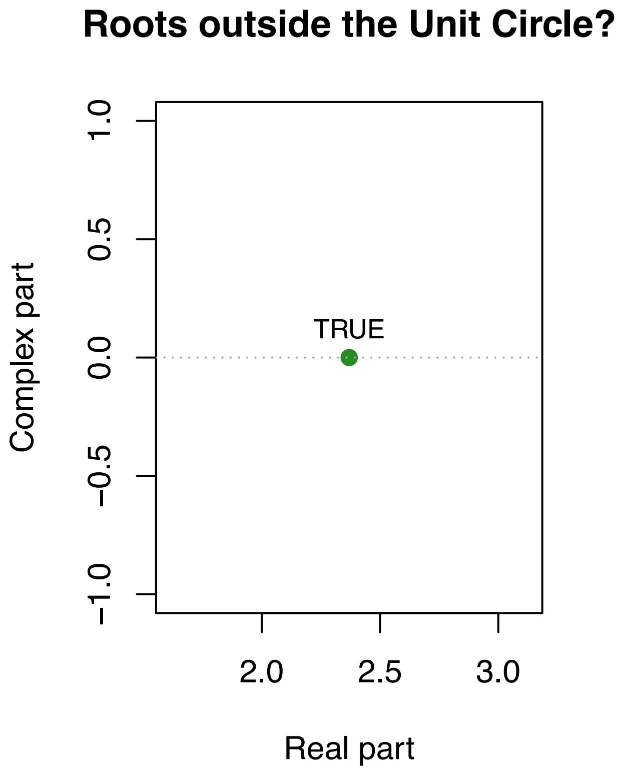
```
##
## Call:
## arima(x = temp.train.BC, order = c(0, 0, 0), seasonal = list(order = c(1, 1,
##     0), period = 12), method = "ML")
##
## Coefficients:
##          sar1
##       -0.4220
## s.e.   0.0774
##
## sigma^2 estimated as 0.07317:  log likelihood = -17.34,  aic = 38.67
```

```
AICc(arima(temp.train.BC,order=c(0,0,0),
                 seasonal=list(order=c(1,1,0),period=12),method="ML"))
```

```
## [1] 38.69932
```

$$(1+0.422B^{12})(1-B^{12})X_t = Z_t$$

```
##       real complex outside
## 1 2.369668       0    TRUE
## *Results are rounded to 6 digits.
```

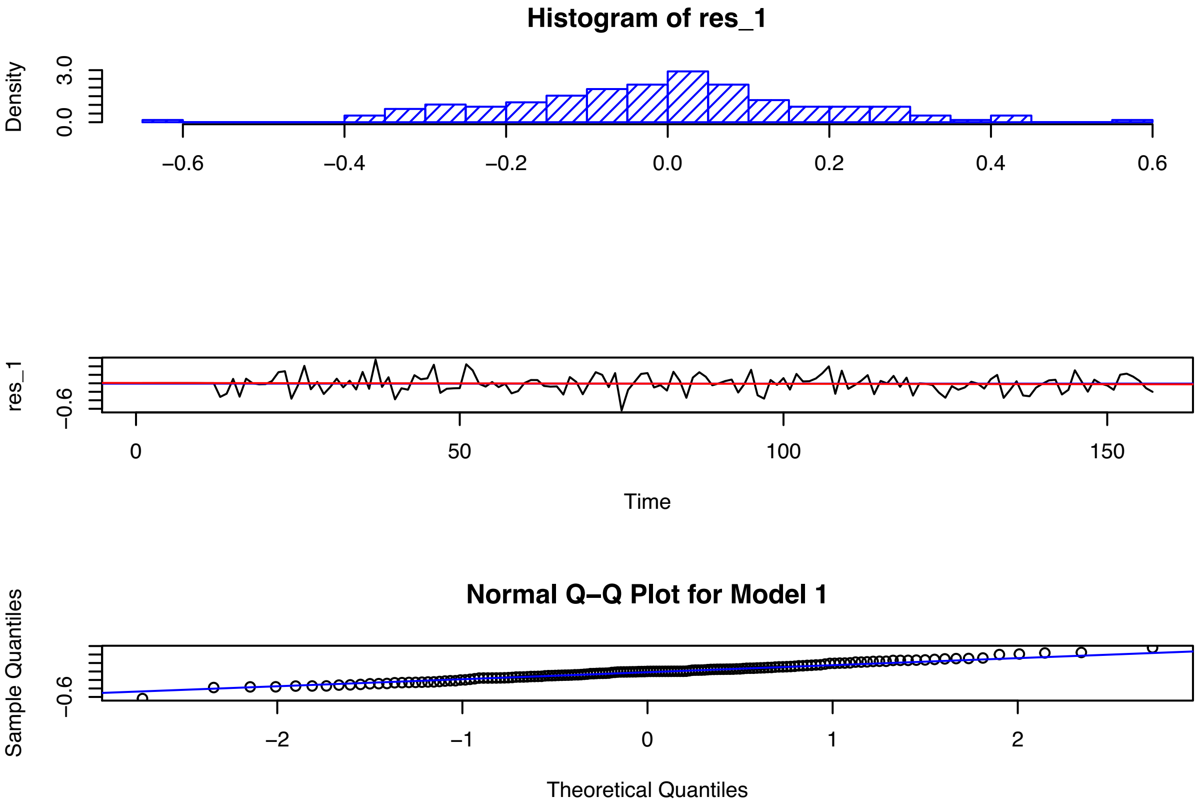


As we can see, by calculating the ploy root of AR part, we find the root is outside the unit circle which imply this model is stationary. And base on this is a pure AR model which is also invertible. So this model is stationary and invertible.
So I'm going to use model 2 and model 4 to process diagnostic checking.

Part 6. Diagnostic Checking

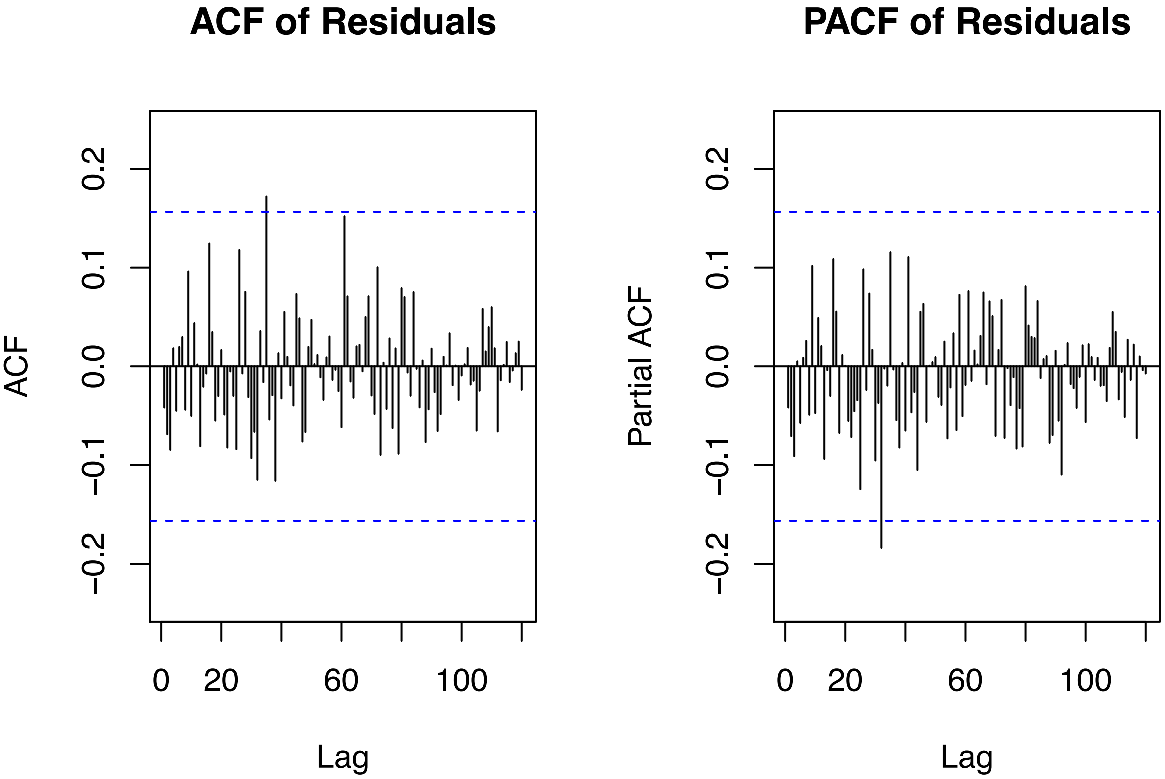
model 2

Residuals, Histogram, and Q-Q Plot for Model 2



The residuals looks pretty good:
1. There aren't trend and have a stable mean.
2. The histogram of the chosen model it is approximately normal.
3. The Q-Q plot is not bad, the most point fall in the regression line.

ACF/PACF of Residuals for Model 2



Although there is a lag in plot ACF and PACF crossing the confidence intervals, but they nearly 0.05 and we can count all the spikes as zero. Thus, the residuals for Model 2 can be thought of as white noise.

Shapiro-Wilk Normality Test, Box-Pierce Test, Box-Ljung Test, McLeod-Li Test; Model 2

```
##
##  Shapiro-Wilk normality test
##
## data:  res_1
## W = 0.99127, p-value = 0.4501
```

```
##
##  Box-Pierce test
##
## data:  res_1
## X-squared = 6.181, df = 9, p-value = 0.7217
```

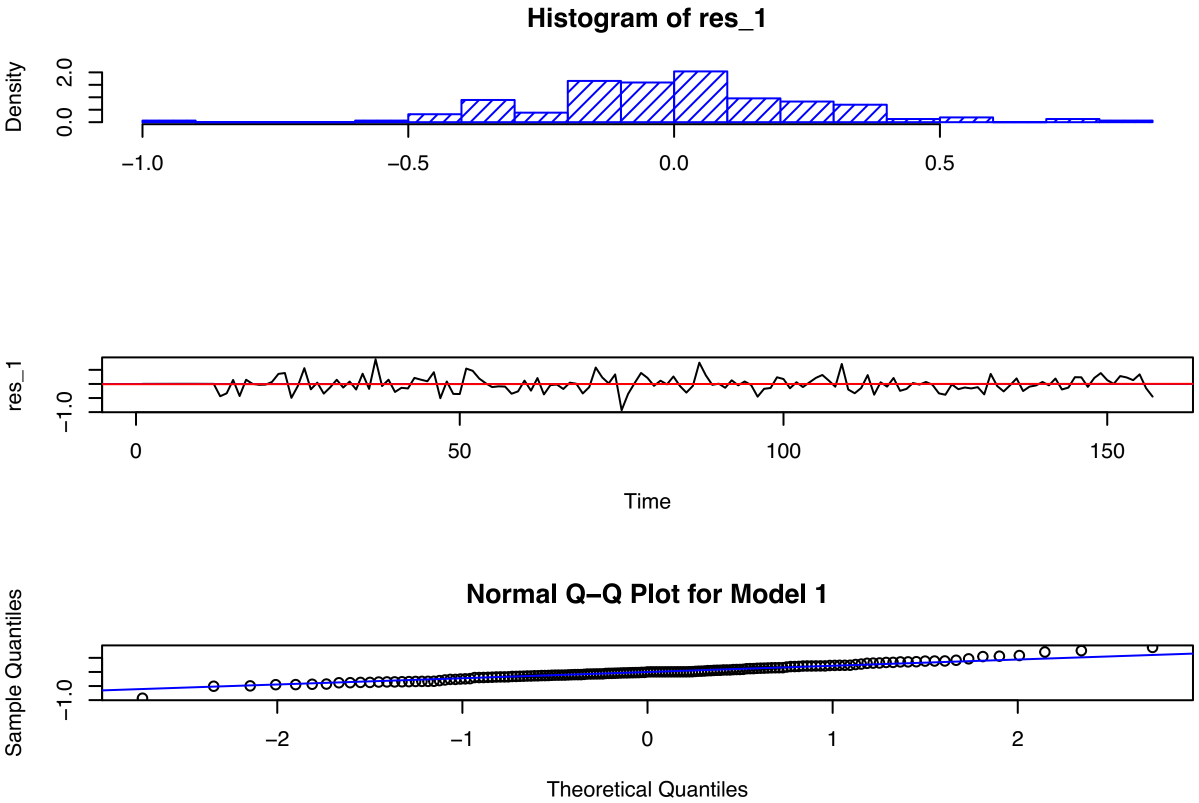
```
##
##  Box-Ljung test
##
## data:  res_1
## X-squared = 6.5641, df = 9, p-value = 0.6824
```

```
##
##  Box-Ljung test
##
## data:  (res_1)^2
## X-squared = 6.5796, df = 13, p-value = 0.9225
```

All of tests pass at the 95% level, the Shapiro-Wilk normality test tells us the residual has normality. The Box-Pierce test tells us that we have zero autocorrelation. The Box-Ljung test, and Box-McLeod test give us similar results.

model 4

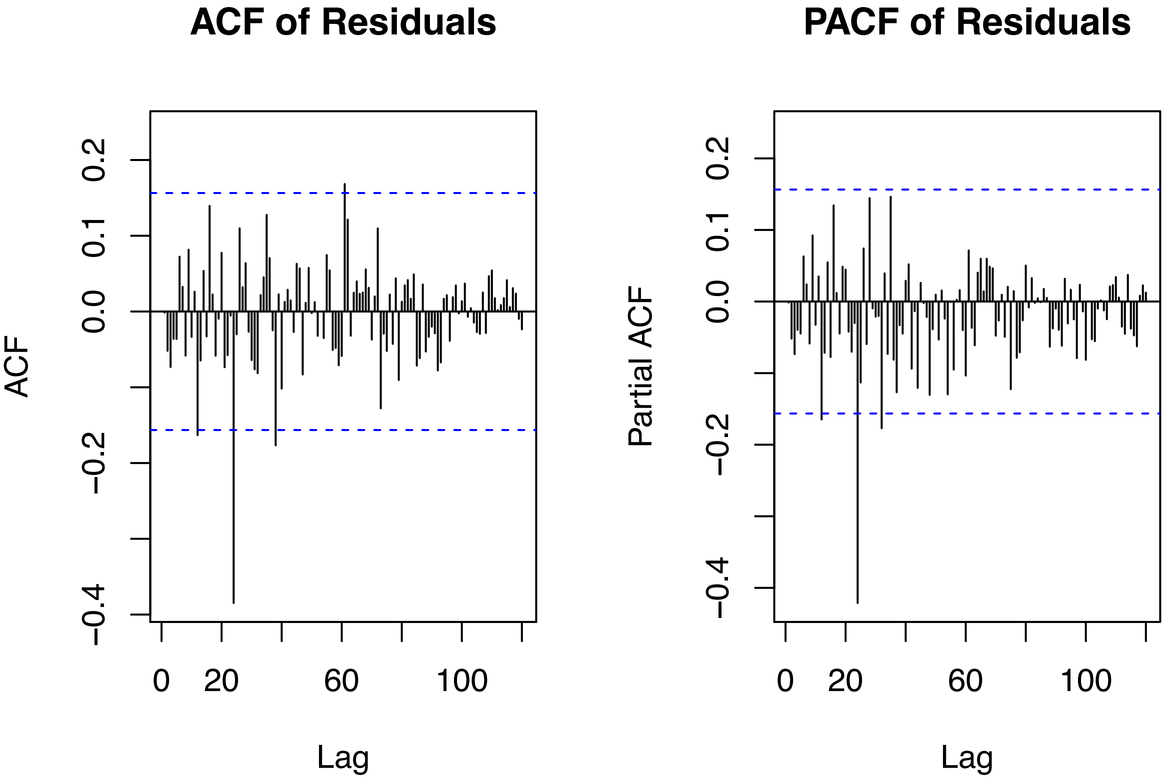
Residuals, Histogram, and Q-Q Plot for Model 4



The residuals looks ok:

1. There aren't trend and have a stable mean.
2. The histogram of the chosen model it is approximately normal but not well by comparing the model 2.
3. The Q-Q plot is not bad, the most points also fall in the regression line.

ACF/PACF of Residuals for Model 4



There are some lags in plot ACF and PACF crossing the confidence intervals, and the lag 24 even reach -0.4.Thus, the residuals for Model 4 may cannot be thought of as white noise.

Shapiro-Wilk Normality Test, Box-Pierce Test, Box-Ljung Test, McLeod-Li Test; Model 4

```
##
##  Shapiro-Wilk normality test
##
## data:  res_1
## W = 0.97929, p-value = 0.01842
```

```
##
##  Box-Pierce test
##
## data:  res_1
## X-squared = 9.3571, df = 12, p-value = 0.6722
```

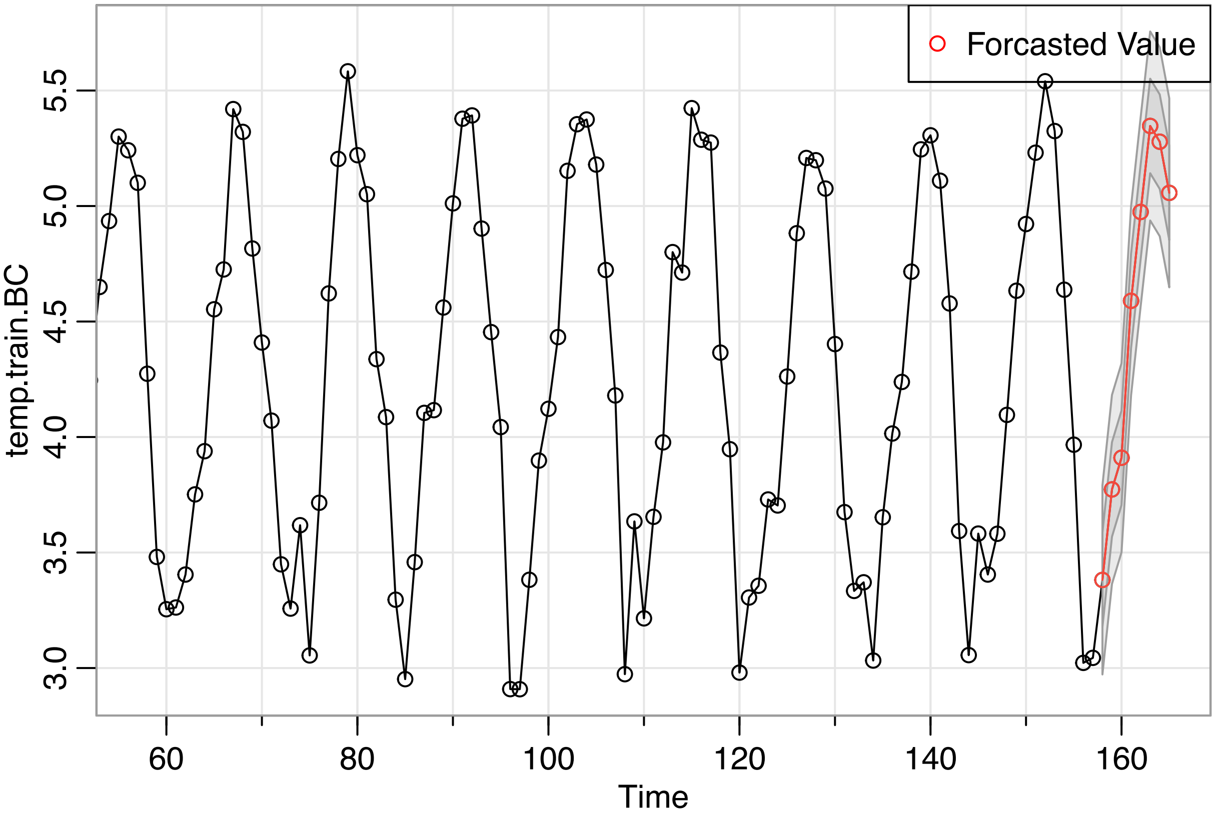
```
##
##  Box-Ljung test
##
## data:  res_1
## X-squared = 10.075, df = 12, p-value = 0.6094
```

```
##
##  Box-Ljung test
##
## data:  (res_1)^2
## X-squared = 14.647, df = 13, p-value = 0.3299
```

All the tests pass at the 95% level, except for the Shapiro-Wilks Normality test which p-value didn't close to the 0.05 level. Hence, by the principle of parsimony, we are going to choose Model 2 as our final model.

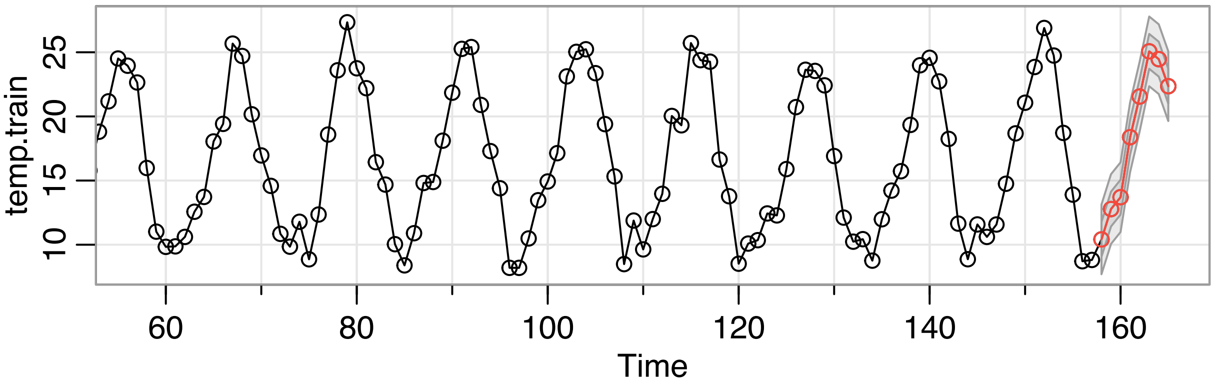
Part 7. Forecasting

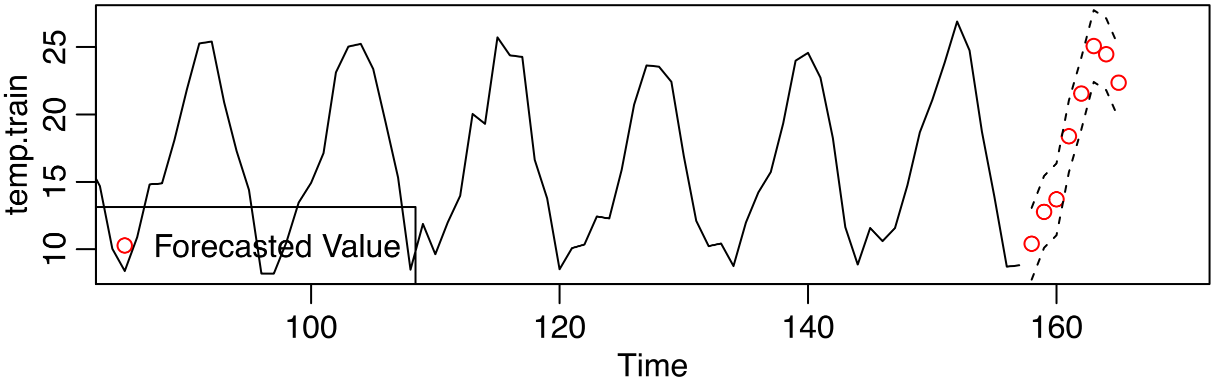
Forecast of Transformed Data Using Model 2



As we can see, the red circles are the forecasted values we produced from the model 2.

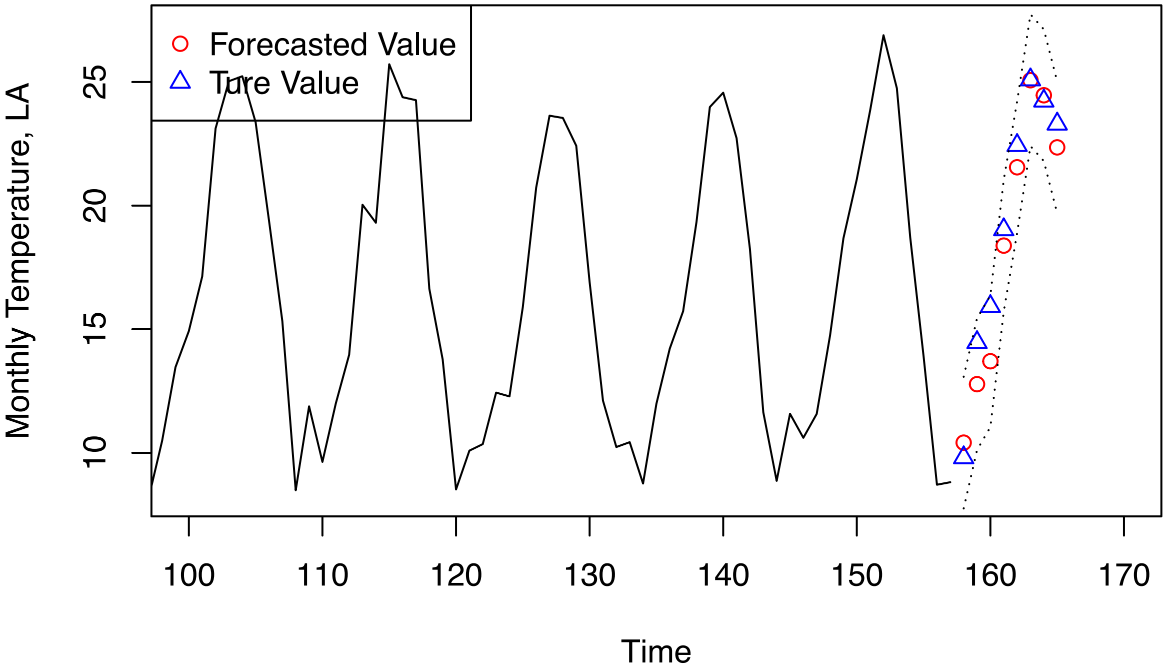
Forecast of Original Data Using Model 2





In order to observe the value better, we can zoom the plot to see the location of each red circles. Since all the red circles fall within the prediction interval, which means the model we construct is reasonable and suitable.

Comaring the Original Data with Forecast Values



As we can see, there have 8 values for each shape that we split at beginning of the project. The blue triangles are the true values and the red circles are the forecasted values. The dashed lines represent the prediction intervals.

Overall, the forecasted values do a pretty good job that all data points locate within the prediction intervals.

Conclusion

My final model:

$$SARIMA(0,0,0)(2,1,2)_{12}$$
$$(1+0.2328B^{24})(1-B^{12})X_t = Z_t$$

The goal of my project was to construct a model that forecasted monthly average temperatures in LA, USA. And applying the time series analysis to a real-world data set.

I believe that I have achieved all goals. Since all the forecasted value have fall within the 95% confident interval and nearly the true value.

References

1. Lecture from PSTAT-174

2. Climate Change: Earth Surface Temperature Data
   https://www.kaggle.com/berkeleyearth/climate-change-earth-surface-temperature-data?select=GlobalLandTemperaturesByMajorCity.csv

3. Detexify LaTex handwritten system
   https://detexify.kirelabs.org/classify.html

Appendix

```
knitr::opts_chunk$set(echo = TRUE)
library(lubridate)
library(dplyr)
library(xts)
library(forecast)
library(ggplot2)
library(qpcR)
library(forecast)
library(MASS)
library(astsa)
library(UnitCircle)
# read data
temp_raw <- read.table("tem_LA.csv", sep=",",
                       header = TRUE, skip = 0)
# ts data
temp <- ts(temp_raw$AverageTemperature,frequency = 12)

# creating training and testing data sets
# leave the last 10 observations for testing
temp.train  <-  temp[c(1:157)]
temp.test  <-  temp[c(158:165)] # 8 obs
# temp.train
plot.ts(temp.train, main='Training Data')
# add trend line and mean line
fit <- lm(temp.train~as.numeric(1:length(temp.train)))
abline(fit,col='red')
abline(h=mean(temp_raw$AverageTemperature), col='blue')
# hist
hist(temp.train)
# plot ACf and PACF
par(mfrow=c(1, 2))
acf(temp.train)
pacf(temp.train)
var(temp.train)
# using Box transformation
bcTransform <- boxcox(temp.train ~ as.numeric(1:length(temp.train)),
                        lambda=seq(-1,3),plotit=TRUE)

lambda <- bcTransform$x[which.max(bcTransform$y)]
lambda

temp.train.BC <- (1/lambda)*(temp.train^lambda-1)

# compare original and box-cox data
par(mfrow=c(1,2))
plot.ts(temp.train,main='Original Data')
ts.plot(temp.train.BC, main = "Box-cox Transform")


# compare the normal distribution
par(mfrow=c(1,2))
```

```r
hist(temp.train, main = 'hist before BC')
hist(temp.train.BC,main = 'hist after BC')

# compare variance
var(temp.train)
var(temp.train.BC)

# decomposition of BC transformed data
temp_new <- ts(temp.train.BC,frequency = 12)
decomp <- decompose(temp_new)
plot(decomp)
# create a table to keep track of variance during differencing
var.table <- matrix(ncol=2,nrow=3)
colnames(var.table) <- c("Differencing","Variance")
var.table <- as.table(var.table)

# No differencing
fit1<-lm(temp.train.BC~as.numeric(1:length(temp.train.BC)));
plot.ts(temp.train.BC)
abline(reg=fit1,col='red')
abline(h=mean(temp.train.BC),col='blue')
# add variance to table
var.table['A','Differencing']= "None"
var.table['A','Variance']=var(temp.train.BC)
var.table
# difference at 12 to remove the seasonality
temp.train.BC_12 <- diff(temp.train.BC,lag=12)
plot.ts(temp.train.BC_12)
fit3<-lm(temp.train.BC_12~as.numeric(1:length(temp.train.BC_12)))
abline(reg=fit3,col='red')
abline(h=mean(temp.train.BC_12),col='blue')
# add variance to table
var.table['B','Differencing']='lag_12'
var.table['B','Variance']=var(temp.train.BC_12,na.rm=TRUE)
var.table
# difference at 1
temp.train.BC_1 <- diff(temp.train.BC_12,lag=1)
plot.ts(temp.train.BC_1)
fit4<-lm(temp.train.BC_1~as.numeric(1:length(temp.train.BC_1)))
abline(reg=fit4,col='red')
abline(h=mean(temp.train.BC_1),col='blue')

# add variance to table
var.table['C','Differencing']='lag12&1'
var.table['C','Variance']=var(temp.train.BC_1,na.rm=TRUE)
var.table
# compare his
par(mfrow=c(1,2))
hist(temp.train.BC)
hist(temp.train.BC_12)

par(mfrow=c(1,2))
Acf(temp.train.BC_12,lag.max=120)
```

```r
Pacf(temp.train.BC_12,lag.max=120)

df <- expand.grid(P=1:2,Q= 0:2 )
df <- cbind(df, AICc=NA)

# Compute AICc:
for (i in 1:nrow(df)) {
  sarima.obj <- NULL
  try(arima.obj <- arima(temp.train.BC, order=c(0, 0, 0),
                        seasonal=list(order=c(df$P[i], 1, df$Q[i]), period=12),
                        method="ML"))
  if (!is.null(arima.obj)) { df$AICc[i] <- AICc(arima.obj) }
  # print(df[i, ])
}
df[order(df$AICc), ]
model_1 <- arima(temp.train.BC,order=c(0,0,0),
                 seasonal=list(order=c(2,1,1),period=12),method="ML")
model_1
par(mfrow=c(1,2))
# check stationarity
polyroot(c(1,-0.2316))
uc.check(pol_ = c(1, -0.2316), plot_output = TRUE)
# check invertibility
uc.check(pol_ = c(1, -1), plot_output = TRUE)
model_2 <- arima(temp.train.BC,order=c(0,0,0),
                 seasonal=list(order=c(2,1,2),period=12),method="ML")
model_2
AICc(arima(temp.train.BC,order=c(0,0,0),
                 seasonal=list(order=c(2,1,2),period=12),method="ML"))

par(mfrow=c(1,2))
# check stationarity
uc.check(pol_ = c(1, -0.2328), plot_output = TRUE)

model_3 <- arima(temp.train.BC,order=c(0,0,0),
                 seasonal=list(order=c(2,1,0),period=12),method="ML")
model_3
AICc(arima(temp.train.BC,order=c(0,0,0),
                 seasonal=list(order=c(2,1,0),period=12),method="ML"))

par(mfrow=c(1,2))
# check stationarity
uc.check(pol_ = c(1, -0.6130,-0.4491), plot_output = TRUE)

model_4 <- arima(temp.train.BC,order=c(0,0,0),
                 seasonal=list(order=c(1,1,0),period=12),method="ML")
model_4
AICc(arima(temp.train.BC,order=c(0,0,0),
                 seasonal=list(order=c(1,1,0),period=12),method="ML"))

par(mfrow=c(1,2))
# check stationarity
uc.check(pol_ = c(1, -0.4220), plot_output = TRUE)
```

```r
par(mfrow=c(3,1))
# plot residuals of Model I
fit_1 <- arima(temp.train.BC,order=c(0,0,0),
                 seasonal=list(order=c(2,1,2),period=12),method="ML")


res_1 <- residuals(fit_1)
m_1 <- mean(res_1)
std_1 <- sqrt(var(res_1))
nt=length(temp_raw$LandAverageTemperature)

# plot histogram of Model 1
hist(res_1,density=20,breaks=20,col='blue',xlab="",prob=TRUE)
plot.ts(res_1)
abline(h=mean(res_1),col='blue')

# plot Q-Q of Model 1 with mean line
fitt_1 <- lm(res_1~as.numeric(1:length(temp.train)))
abline(fitt_1,col= 'red')
qqnorm(res_1,main="Normal Q-Q Plot for Model 1")
qqline(res_1,col="blue")
# plot ACF and PACF of residuals for Model I, side by side
par(mfrow=c(1,2))
Acf(res_1,lag.max=120,main="ACF of Residuals")
Pacf(res_1,lag.max=120,main="PACF of Residuals")
# Shapiro-Wilk Normality Test, Box-Pierce Test, Box-Ljung Test, McLeod-Li Test; Model I
shapiro.test(res_1)
Box.test(res_1,lag=13,type=c("Box-Pierce"),fitdf=4)
Box.test(res_1,lag=13,type=c("Ljung-Box"), fitdf=4)
Box.test((res_1)^2,lag=13,type=c("Ljung-Box"),fitdf=0)
par(mfrow=c(3,1))
# plot residuals of Model I
fit_1 <- arima(temp.train.BC,order=c(0,0,0),
                 seasonal=list(order=c(1,1,0),period=12),method="ML")


res_1 <- residuals(fit_1)
m_1 <- mean(res_1)
std_1 <- sqrt(var(res_1))
nt=length(temp_raw$LandAverageTemperature)

# plot histogram of Model 1
hist(res_1,density=20,breaks=20,col='blue',xlab="",prob=TRUE)
plot.ts(res_1)
abline(h=mean(res_1),col='blue')

# plot Q-Q of Model 1 with mean line
fitt_1 <- lm(res_1~as.numeric(1:length(temp.train)))
abline(fitt_1,col= 'red')
qqnorm(res_1,main="Normal Q-Q Plot for Model 1")
qqline(res_1,col="blue")
# plot ACF and PACF of residuals for Model I, side by side
par(mfrow=c(1,2))
```

```r
Acf(res_1,lag.max=120,main="ACF of Residuals")
Pacf(res_1,lag.max=120,main="PACF of Residuals")
# Shapiro-Wilk Normality Test, Box-Pierce Test, Box-Ljung Test, McLeod-Li Test; Model I
shapiro.test(res_1)
Box.test(res_1,lag=13,type=c("Box-Pierce"),fitdf=1)
Box.test(res_1,lag=13,type=c("Ljung-Box"), fitdf=1)
Box.test((res_1)^2,lag=13,type=c("Ljung-Box"),fitdf=0)
# forecast of Transformed Data Using Model 2
mypred.bc = sarima.for(temp.train.BC,n.ahead=8,
                    p=0,d=0,q=0,P=2,D=1,Q=2,S=12,
                    no.constant = FALSE, plot.all=F)
# add legend to plot
legend('topright',pch=1,col = c('red'),
       legend = c('Forcasted Value'))

# plot zoomed in version of forecast of original data
par(mfrow=c(2,1))
mypred.og = sarima.for(temp.train,n.ahead=8,
                    p=0,d=0,q=0,P=2,D=1,Q=2,S=12,
                    no.constant = FALSE, plot.all=F)

ts.plot(temp.train,xlim=c(86,length(temp.train)+12))
# add points and lines based on prediction
points(158:165, mypred.og$pred,col= 'red')
lines(158:165, (mypred.og$pred+1.96*mypred.og$se),lty=2)
lines(158:165, (mypred.og$pred-1.96*mypred.og$se),lty=2)
legend('bottomleft',fill,pch=1,col = c('red'),
       legend = c('Forecasted Value'))

# forecast of original data with true values on top
ts.plot(temp.train,xlim=c(100,170),ylab= 'Monthly Temperature, LA')

# add forecasted values
points(158:165,mypred.og$pred,col= 'red')

# add true values
points(158:165,temp.test,col="blue",pch=2)

# add legend and prediction intervals
legend("topleft",pch=c(1,2),fill,col=c('red','blue'),
         legend=c('Forecasted Value','Ture Value'))
lines(158:165,(mypred.og$pred+1.96*mypred.og$se),lty=3)
lines(158:165,(mypred.og$pred-1.96*mypred.og$se),lty=3)
```